CLAIMS

Please amend the claims as follows:

1. (currently amended) A system for encrypting/decrypting messages, comprising:

a public key cryptosystem <u>further comprising a computer operable for generating keys for use with messages that have been encrypted and/or decrypted wherein the public key cryptosystem</u> having a predetermined number of prime factors used for the generation of a modulus N and an exponent e;

wherein a proper subset of the prime factors of the modulus N, along with the exponent e, are required to decrypt messages that are encrypted using the public exponent e and the public modulus N, where e and N are calculated using RSA methods, and encryption occurs using RSA methods.

2. (currently amended) A method for encrypting/decrypting messages comprising the steps of: providing a public key cryptosystem <u>including a computer operable to generate at least one key for encrypting/decrypting at least one message, the public key cryptosystem</u> having a predetermined number of prime factors used for the generation of a modulus N and an exponent e;

wherein a proper subset of the prime factors of the modulus N are required to decrypt messages that are encrypted using the public exponent e and the public modulus N, where e and N are calculated using RSA methods, and encryption <u>of the message</u> occurs using RSA methods.

3. (currently amended) A method for encrypting/decrypting messages comprising the steps of: Encrypting <u>on a computer</u> a plaintext message M into a ciphertext message C using any method that produces a value equivalent to $C = M^e \bmod N$, where $0 \leq M < N_d$, such that the ciphertext C can be decrypted into the plaintext message M using only e and the prime factors of N_d

N being the product of all of the numbers in the set S;

S being a set of at least two prime numbers, $p_1 \ldots p_k$, where k is an integer greater than 1;

e being a number;

S_d being a proper subset of S;

N_d being the product of all of the numbers in the set S_d.

4. (original) The method of claim 3, wherein the step of generating the exponent e includes calculating the exponent e as a number that is relatively prime to the product of each distinct prime factor of N minus 1, $(N_1 - 1)^* \ldots (N_j - 1)$ for distinct prime factors of N 1 to j, where j is the number of distinct prime factors in N, or choosing the exponent e as a small prime number.

5. (currently amended) A method for decrypting encrypted messages comprising the steps of:

determining if a derived modulus N_d is a squarefree number, and if so,

decrypting <u>on a computer</u> ciphertext C into message M <u>wherein message M was originally an encrypted message that is transformed into electronic, decrypted message M</u> using any method that produces a value equivalent to $M = C^d \bmod N_d$, where d is generated using the following steps:

calculating the number Z_d as the product of each prime factor of N_d minus 1, $(N_{d1} - 1)^* \ldots (N_{dj} - 1)$ for prime factors of N_d 1 to j, where j is the number of prime factors in N_d;

generating the exponent d such that the following relationship is satisfied: $e^*d = 1 \bmod Z_d$.

6. (original) The method according to claim 5, further including the step of:
directly calculating $M = C^d \bmod N_d$.

7. (original) The method according to claim 5, further including the steps of:

calculating separate decryption exponents $d_{nd1}...d_{ndj}$ for all prime factors of N_d 1 to j, where j is the number of prime factors in N_d so that the following relationship is satisfied for each member of N_d: $e*d_{ndi} = 1 \bmod (N_{di} - 1)$; and

performing decryptions of the form $M_i = C^{d_{ndi}} \bmod N_{di}$ for all prime factors of N_d from 1 to j, where j is the number of prime factors in N_d, and then using the values of each M_i and N_{di} to reconstruct M.

8. (original) The method of claim 7, wherein the values of each M_i and N_{di} restore the plaintext message M using the Chinese Remainder Theorem and/or Garner's algorithm.

9. (currently amended) A method for decrypting encrypted messages, comprising the steps of:

decrypting <u>on a computer</u> the ciphertext message C to the plaintext message M by determining if the derived modulus N_d is squareful number, and if so;

calculating separate decryption exponents $d_{nd1}...d_{ndj}$ for all distinct prime factors of N_d 1 to j, where j is the number of distinct prime factors in N_d so that the following relationship is satisfied for each distinct member of N_d: $e*d_{ndi} = 1 \bmod (N_{di} - 1)$;

for each distinct prime factor of N_d, N_{di}, calculating a value b_{di} as the number of times that N_{di} occurs as a prime factor in N_d;

calculating M_i for each distinct prime factor of N_d, N_{di};

and using all values of M_i, N_{di}, d_{ndi}, and b_{di} to <u>transform the plaintext message M and to</u> restore the plaintext message M <u>from an encrypted to a decrypted form</u>.

10. (original) The method of claim 9, further including the steps of:

using Hensel Lifting to calculate M_i for each distinct prime factor of N_d, N_{di}.

11. (original) The method of claim 9, further including using techniques such as the Chinese Remainder Theorem and/or Garner's algorithm to use all value of M_i, $N_{di,}$ d_{ndi}, and b_{di} to restore the plaintext message M.

12. (currently amended) A public key cryptosystem where messages are decrypted <u>on a computer</u> using a set of prime numbers S and the public exponent e, and messages are encrypted using a modulus N_p that is calculated as the product of a set of numbers that is a proper superset of S, and encryption occurs with standard RSA methods using the public exponent e and the modulus N_p.

13. (currently amended) A method for encrypting/decrypting messages, comprising the steps of:

Encrypting <u>on a computer</u> a plaintext message M into a ciphertext message C using any method that produces a value equivalent to $C = M^e \bmod N_p$, where $0 \leq M < N$, such that the ciphertext C can be decrypted into the plaintext message M using e and the prime factors of N

N being the product of all of the numbers in the set S;

S being a set of at least one prime number, $p_1 \ldots p_k$, where k is an integer greater than 0;

S_p being a proper superset of S;

N_p being the product of all of the numbers in the set S_p;

e being a number.

14. (original) The method of claim 13, wherein the step of generating the exponent e includes calculating the exponent e as a number that is relatively prime to the product of each distinct prime factor of N_p minus 1, $(N_{p1} - 1)^* \ldots (N_{pj} - 1)$ for distinct prime factors of N_p 1 to j, where j is the number of distinct prime factors in N_p.

15. (original) The method of claim 13, wherein the step of generating the exponent e includes choosing the exponent e as a small prime number.

16. (currently amended) A method for decrypting encrypted messages, including the steps of:

Decrypting on a computer the ciphertext message C to the plaintext message M by:

determining if the derived modulus N is squareful number; if so then, calculating separate decryption exponents $d_{n1} \ldots d_{nj}$ for all distinct prime factors of N 1 to j, where j is the number of distinct prime factors in N so that the following relationship is satisfied for each distinct member of N: $e^*d_{ni} = 1 \bmod (N_i - 1)$;

for each distinct prime factor of N, N_i, calculating a value b_i as the number of times that N_i occurs as a prime factor in N;

calculating M_i for each distinct prime factors of N, N_i;

and using each value of Mi, Ni, bi and dni to restore the plaintext message ~~M;~~ M.

17. (original) The method of claim 16, where Hensel Lifting is used to calculate Mi for each distinct prime factor of N, Ni.

18. (original) The method of claim 16, further including using techniques such as the Chinese Remainder Theorem and/or Garner's algorithm to use all value of Mi, Ni, dni, and bi to restore the plaintext message M.

19. (currently amended) A method of decrypting encrypted messages, including the steps of:

Decrypting on a computer the ciphertext message C into the plaintext message M by:

determining if the modulus N is a squarefree number; and if so then,

decrypting ciphertext C into message M using any method that produces a value equivalent to M = Cd mod N, where d is generated using the following steps:

Calculating the number Z as the product of each prime factor of N minus 1, (N1 – 1)*...(Nj – 1) for prime factors of N 1 to j, where j is the number of prime factors in N;

then generating the decryption exponent d such that the following relationship is satisfied: e*d = 1 mod Z.

20. (original) The method according to claim 19, further including the step of: directly calculating M = Cd mod N.

21. (original) The method according to claim 19, further including the steps of:

calculating separate decryption exponents d1...dj for all prime factors of N 1 to j, where j is the number of prime factors in N so that the following relationship is satisfied for each member of N: e*di = 1 mod (Ni – 1); and performing decryptions of the form Mi = Cdi mod Ni for all prime factors of N from 1 to j, where j is the number of prime factors in N, and then using the values of each Mi and Ni to reconstruct M.

22. (original) The method of claim 21, wherein the values of each Mi and Ni reconstruct M using the Chinese Remainder Theorem and/or Garner's algorithm.

23. (currently amended) A method for encrypting/decrypting messages comprising the steps of:

Encrypting <u>on a computer</u> a plaintext message M into a ciphertext message C using any method that produces a value equivalent to C = Me mod Np, where 0 ≤ M < N, such that the ciphertext C can be decrypted into the plaintext message M using e and the prime factors of N.

N being the product of all of the members of set S;

S being a set of at least two numbers, p1...pk where k is an integer greater than 1 and all members of S are equal to ps, which is a prime number;

Sp being a superset of S;

Np being the product of all of the numbers in the set Sp;

e being a number.

24. (original) The method of claim 23, wherein the step of generating the exponent e further includes: Calculating the exponent e as a number that is relatively prime to the product of all of the distinct prime factors of Np minus 1, (Np1 – 1)*...(Npj – 1) for distinct prime factors of Np 1 to j, where j is the number of distinct prime factors in Np.

25. (original) The method of claim 23, wherein the step of generating the exponent e includes choosing the exponent e as a small prime number.

26. (currently amended) A method of decrypting encrypted messages, including the steps of:

Decrypting <u>on a computer</u> the ciphertext message C to the plaintext message M by:

Calculating b as the number of times that the number ps occurs as a prime factor in N;

Generating an exponent d such that the following equation is satisfied:

e*d = 1 mod (ps – 1);

Using Hensel Lifting to transform C into M with d, ps, and b as input values.

27. (currently amended) A method for encrypting/decrypting messages, comprising the steps of:

Encrypting <u>on a computer</u> a plaintext message M into a ciphertext message C using any method that produces a value equivalent to C = Me mod Np, where $0 \leq M < p$, such that the ciphertext C can be decrypted into the plaintext message M using e and p

p being a prime number;

S being a set containing only the number p;

Sp being a superset of S;

Np being the product of all members of the set Sp;

e being a number.

28. (original) The method of claim 27, wherein the step of generating the exponent e further includes: Calculating the exponent e as a number that is relatively prime to the product of each distinct prime factor of Np minus 1, (Np1 – 1)*...(Npj – 1) for distinct prime factors of Np 1 to j, where j is the number of distinct prime factors in Np.

29. (original) The method of claim 27, wherein the step of generating the exponent e includes choosing the exponent e as a small prime number.

30. (currently amended) A method for decrypting encrypted messages, comprising the steps of:

Decrypting <u>on a computer</u> using any method that produces a value equivalent to as M = Cd mod p, where d is generated using the following step:

Calculating d such that the following equation is satisfied:

e*d = 1 mod (p – 1).

31. (currently amended) A method for establishing cryptographic communications, comprising the steps of:

calculating a composite number N, which is formed from the product of distinct prime numbers S, p1,...pk where k ≥ 1.

<u>on a computer</u> Encoding a plaintext message M, to a ciphertext C, where M corresponds to a number representative of a message and 0 ≤ M < S;

generating an exponent e;

transforming <u>on the computer</u> said plaintext, M, into said ciphertext, C, where C is developed using any method that produces a value equivalent to C = Me mod N, such that ciphertext C can be decrypted into plaintext M using only e and S.

32. (original) The method of claim 31, wherein the step of generating the exponent e further includes: Calculating the exponent e as a number that is relatively prime to the product of each distinct prime factor of N minus 1, (N1 – 1),...(Nj – 1) for distinct prime factors of N 1 to j, where j is the number of distinct prime factors in N.

33. (original) The method of claim 31, wherein the step of generating the exponent e includes choosing the exponent e as a small prime number.

34. (currently amended) A method for decrypting encrypted messages, comprising the steps of:

decoding <u>on a computer</u> the ciphertext message C to the plaintext message M, wherein said decoding comprises the step of: transforming said ciphertext message C to plaintext M, using any method that produces a value equivalent to M = Cd mod S, where d is generated using the following step:

generating d such that e*d = 1 mod (S – 1).

35. (original) A system for encrypting and decrypting electronic communications including a network of computers and/or computer-type devices, such as personal data assistants (PDAs), mobile phones and other devices, in particular mobile devices capable of communicating on the network; generating at least one private key and at least one public key, wherein the at least one private key is determined based upon any one of a multiplicity of prime numbers that when multiplied together produce N, which is the modulus for at least one of the public keys.

36. (currently amended) A method for public key decryption where less than all of the distinct prime factors of a number N are used to decrypt a ciphertext message C into plaintext message M, where encryption occurs <u>on a computer</u> with the public key {e, N} using any method that produces a value equivalent to C = Me mod N.

37. (currently amended) A method for public key encryption with a public key {e, N} where a plaintext message M is encrypted <u>on a computer</u> into a ciphertext message C using any method that produces a value equivalent to C = Me mod (N*X), where N is the public modulus and X is any integer greater than 1.

38. (currently amended) A method for public key decryption of a message that has been encrypted with the public key {e, N} where a ciphertext message C is decrypted <u>on a computer</u> into a plaintext message M using any method that produces a value equivalent to M = Cd mod Nd, where Nd is the product of less than all of the prime factors of the public modulus N and d satisfies the equation e*d = 1 mod Z, where Z is the product of each of the k prime factors of Nd minus 1, (p1 – 1)*...(pk – 1).

39. (currently amended) A method for public key decryption of a message that has been encrypted <u>on a computer</u> using any method that produces a value equivalent to C = Me mod N, where a ciphertext message C is decrypted into a plaintext message M using any method that produces a value equivalent to M = Cd mod Nd, where Nd is the product of less than all of the prime factors of the public modulus N and d satisfies the equation e*d = 1 mod Z, where Z is the product of each of the k prime factors of Nd minus 1, (p1 – 1)*...(pk – 1).